

MI PRIMER PROYECTO DE MACHINE LEARNING

umaker_

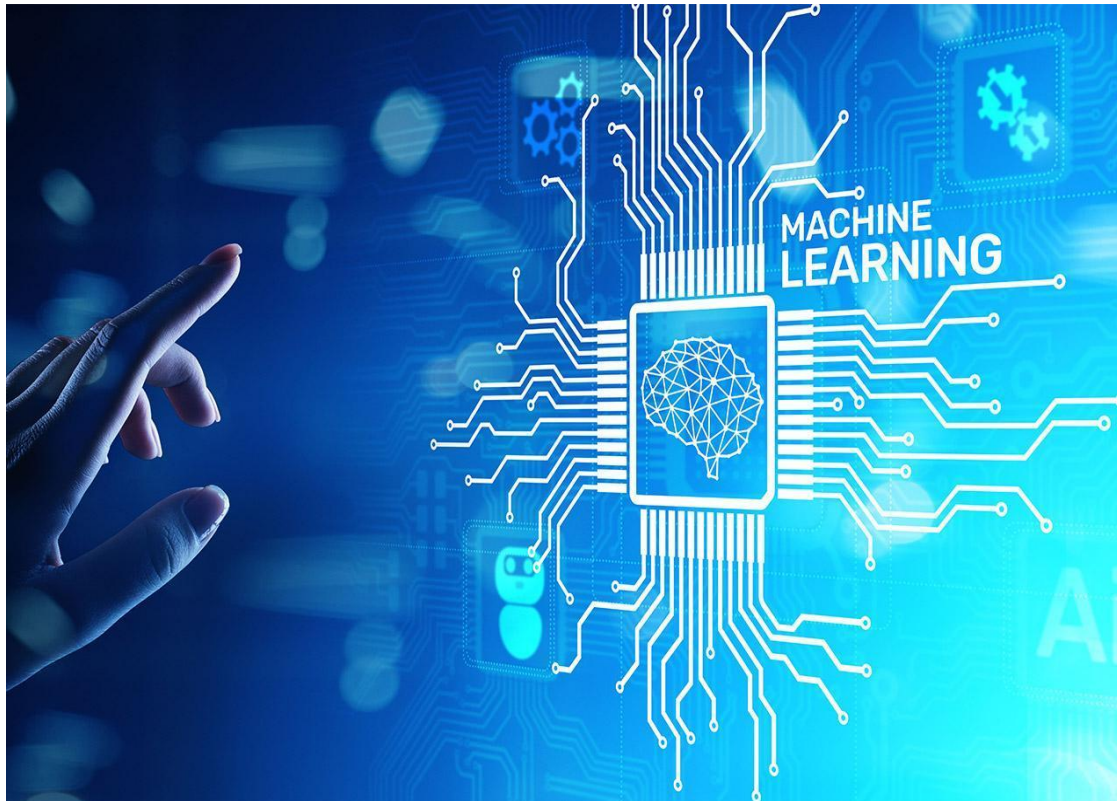
 CODAERUS



JKSOLTEC
Soluciones IOT

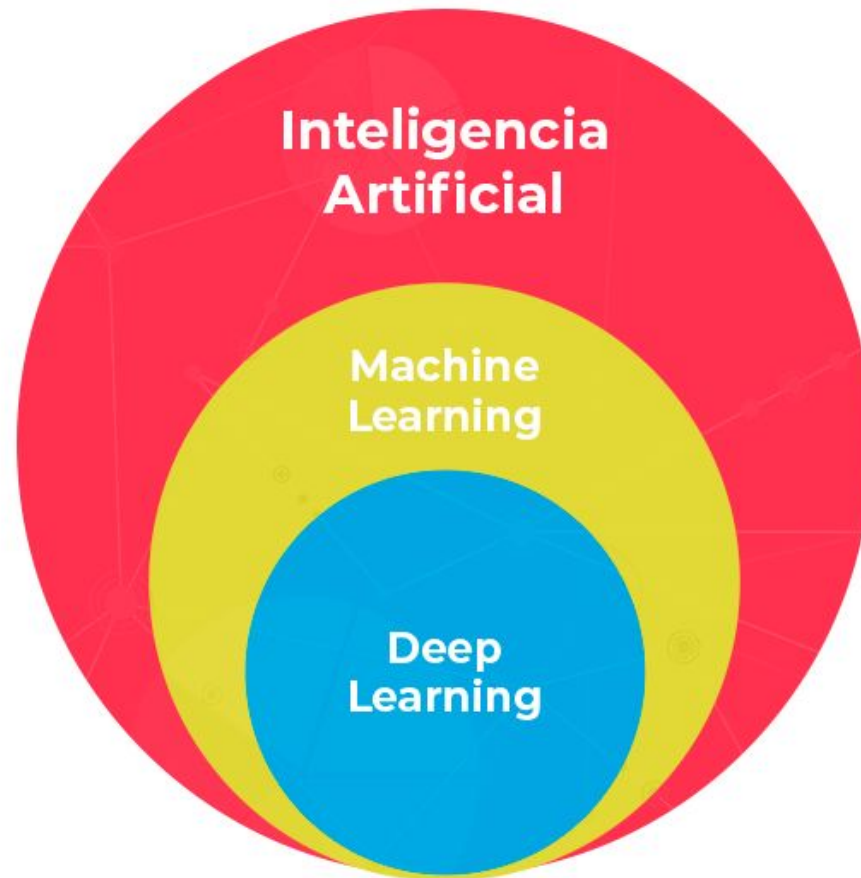
INTELIGENCIA ARTIFICIAL

La **IA** (Inteligencia Artificial) es una rama de la informática que se centra en la creación de sistemas y programas capaces de realizar tareas que normalmente requieren inteligencia humana.



Relación entre IA, ML y DL

La inteligencia artificial es un campo amplio que incluye subcampos como el Machine Learning y el Deep Learning.



EN NUESTRO DIA A DIA

El machine learning está presente en numerosos aspectos de nuestra vida cotidiana, a menudo de manera invisible pero significativa.

INTERNET

REDES
SOCIALES

ASISTENTES

PUBLICIDAD

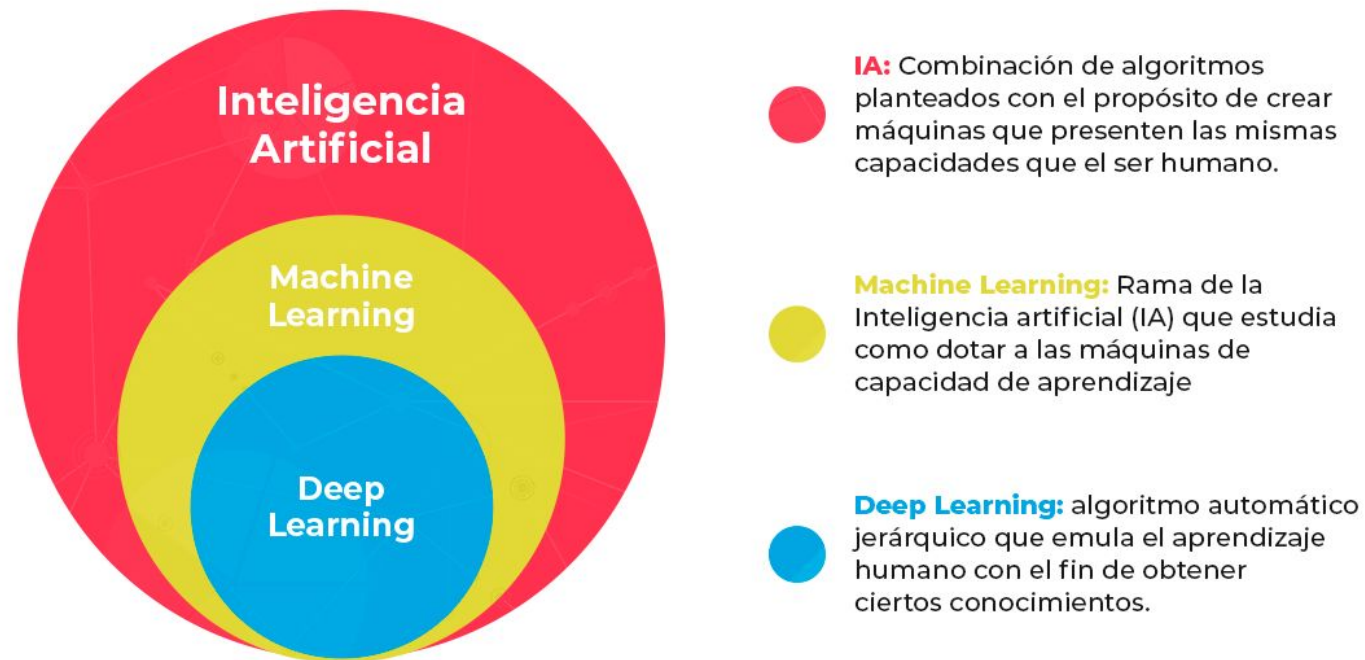
FRAUDES

MEDICINA

TRANSPORTE

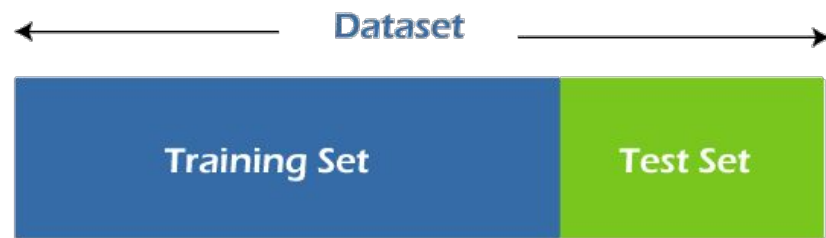
Redes Neuronales Artificiales (RNA)

- Las Redes Neuronales Artificiales se corresponden con el componente principal del Deep Learning.
- Se introducen por primera vez en 1943 por el neuropsicólogo Warren McCulloch y el matemático Walter Pitts.
- En la década de los 1990s pierden interés a favor de técnicas de Machine Learning como el Support Vector Machine.
- En 2012 el Deep Learning resurge con más fuerza que nunca cuando Geoffrey Hinton gana el reto ImageNet [2] con una Red Neuronal Convolutacional.

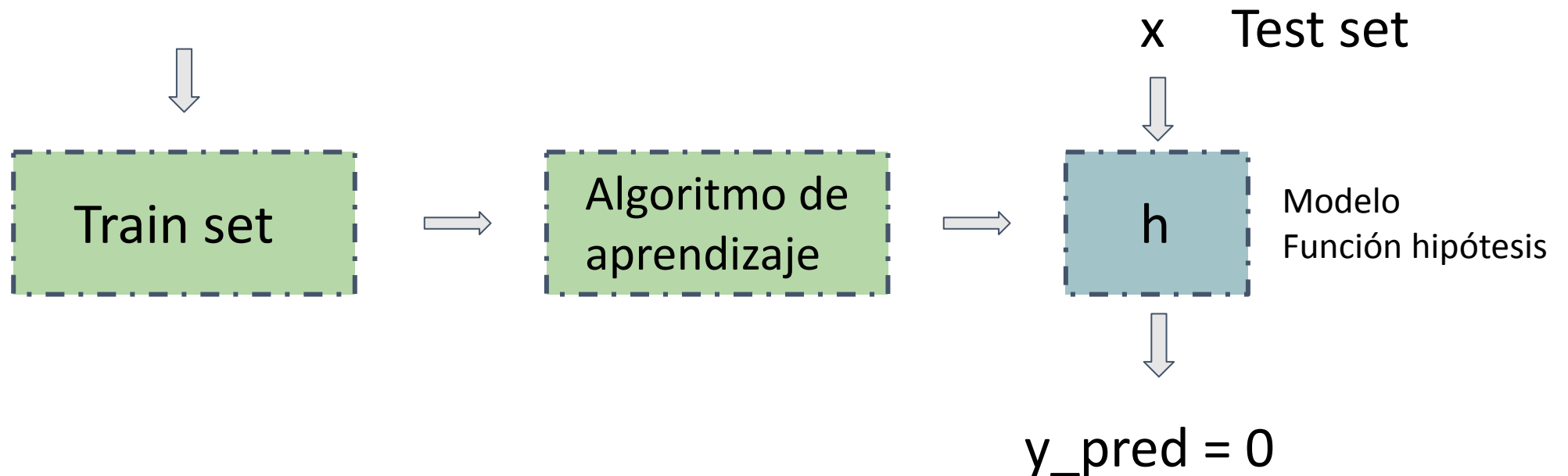


MACHINE LEARNING

Es una rama de la IA permite a los sistemas aprender patrones sin ser programados explícitamente.

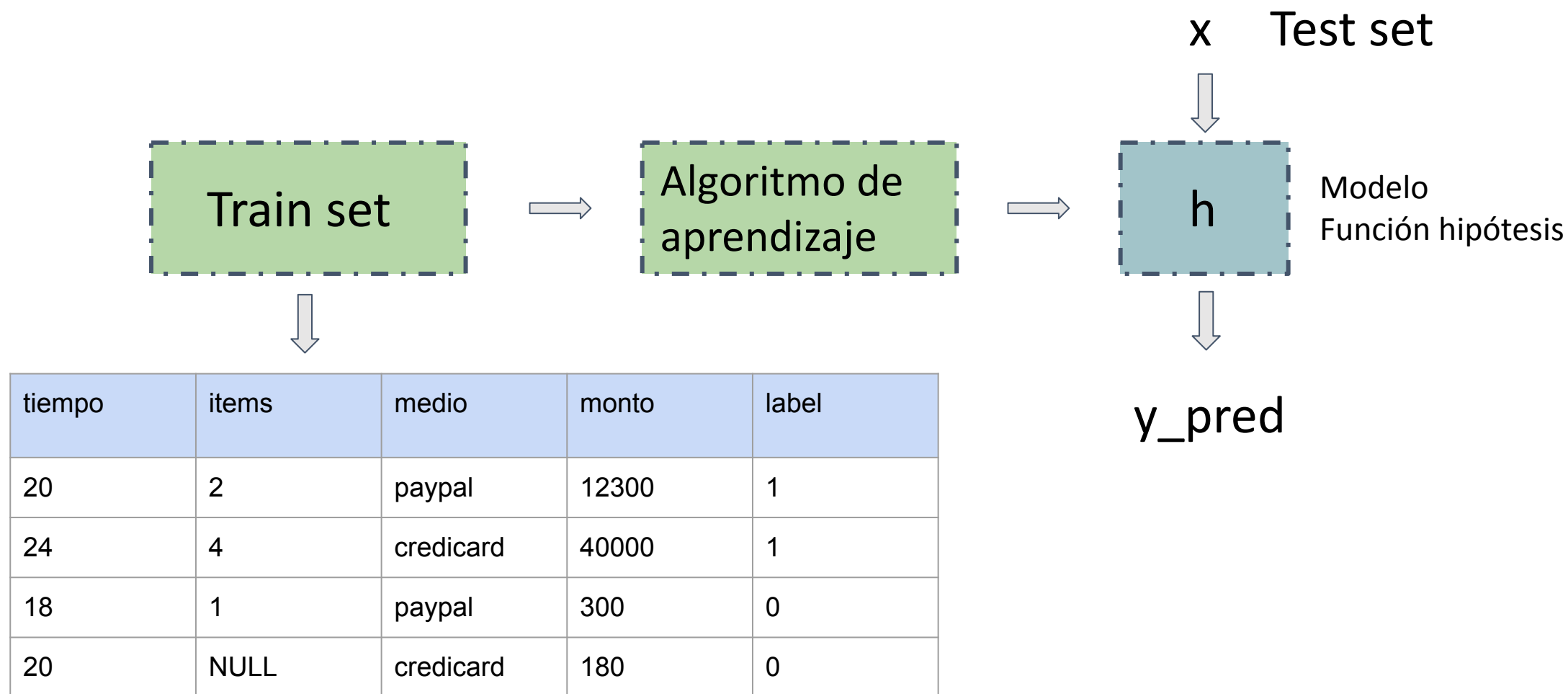


| tiempo | items | medio | monto | label |
|--------|-------|-----------|-------|-------|
| 20 | 2 | paypal | 12300 | 1 |
| 24 | 4 | credicard | 40000 | 1 |
| 18 | 1 | paypal | 300 | 0 |
| 20 | NULL | credicard | 0.09 | 0 |



MACHINE LEARNING

Es una rama de la IA permite a los sistemas aprender patrones sin ser programados explícitamente.



RUTA DE APRENDIZAJE PARA INTELIGENCIA ARTIFICIAL

PROCESAMIENTO DIGITAL DE IMÁGENES

Técnicas para analizar, mejorar y extraer información de imágenes con herramientas como OpenCV y PIL.

MACHINE LEARNING

Algoritmos que permiten a los sistemas aprender de datos y predecir resultados sin programación explícita.

NLP

Aprende a procesar texto con NLP, redes neuronales y machine learning para crear aplicaciones que entienden y generan lenguaje humano.

IA GENERATIVA

Modelos avanzados (GANs, transformers) capaces de generar texto, imágenes, audio y más a partir de datos existentes.

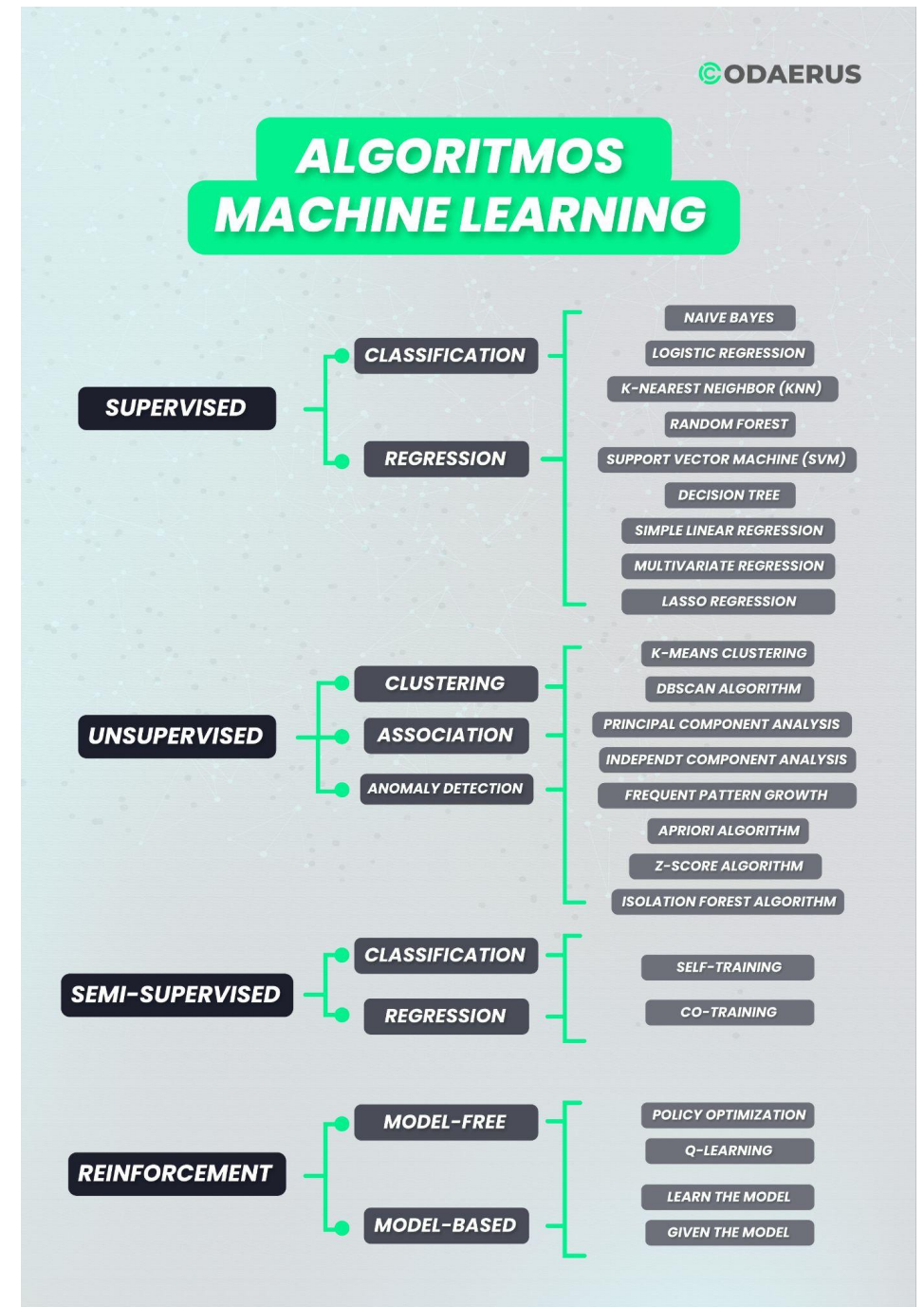
PYTHON

Lenguaje esencial en IA por su sintaxis sencilla y librerías poderosas como NumPy, Pandas, OpenCV, TensorFlow y Keras.

DEEP LEARNING

Redes neuronales profundas que resuelven tareas complejas como visión por computador y procesamiento del lenguaje.

Enfoques específicos para resolver distintos problemas. Para entender y aplicar estos métodos, es clave dominar matemáticas como derivadas, gradientes, álgebra lineal, integrales, probabilidad, estadística, distribución gaussiana y series de Taylor. Estos fundamentos permiten analizar, optimizar y ajustar modelos con precisión. Además, conocer Python y sus librerías es esencial para implementar soluciones efectivas en la práctica.



EL PROBLEMA DE LAS CARACTERÍSTICAS

El rendimiento de estos algoritmos de Machine Learning depende en gran medida de la representación de los datos que se proporciona.

Muchas tareas pueden resolverse diseñando un conjunto adecuado de características y proporcionándoles a un algoritmo de Machine Learning.

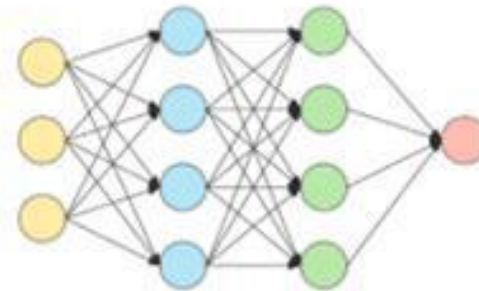
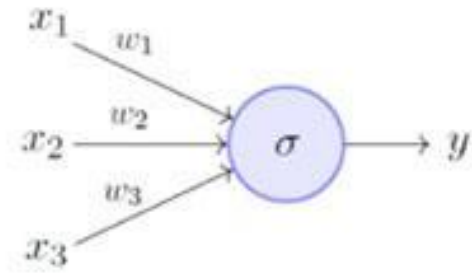
Sin embargo, para otras tareas, es difícil identificar qué características deben ser extraídas, por ejemplo: Imágenes, audio, etc.



¿Por qué resurgen las RNA?

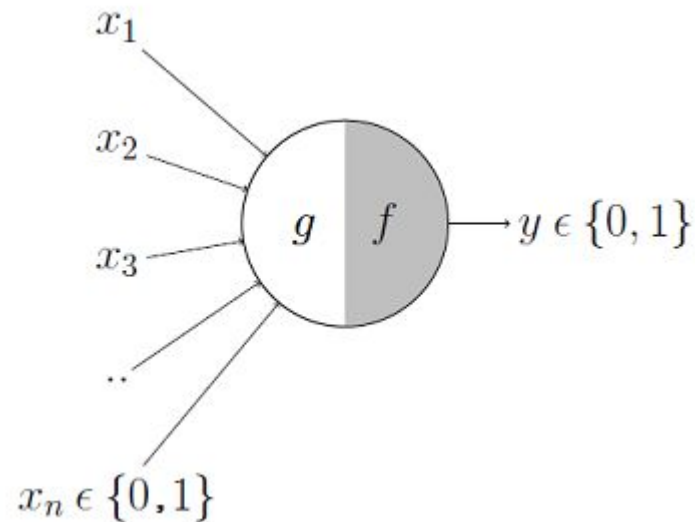
- En la actualidad existe una gran cantidad de datos disponibles.
- En las últimas décadas el poder computacional se ha multiplicado, permitiendo la ejecución de algoritmos costosos y complejos en períodos de tiempo razonables.
- Los algoritmos que implementan las RNA han mejorado.
- Muchas limitaciones que se habían intuido de manera teórica resultaron ser benignas en la práctica.

NEURONAS BIOLÓGICAS Y ARTIFICIALES



Neurona de McCulloch y Pitts (Neurona M-P)

- Se corresponde con la primera neurona artificial de la historia.
- Se caracteriza porque recibe uno o más valores binarios $\{1,0\}$ y retorna otro valor binario $\{1,0\}$.
- Activa su salida cuando más de un número determinado de valores de entrada se encuentran activos.
- Debe establecerse manualmente el número de valores de entrada que deben estar activos, a este valor se le denomina threshold.

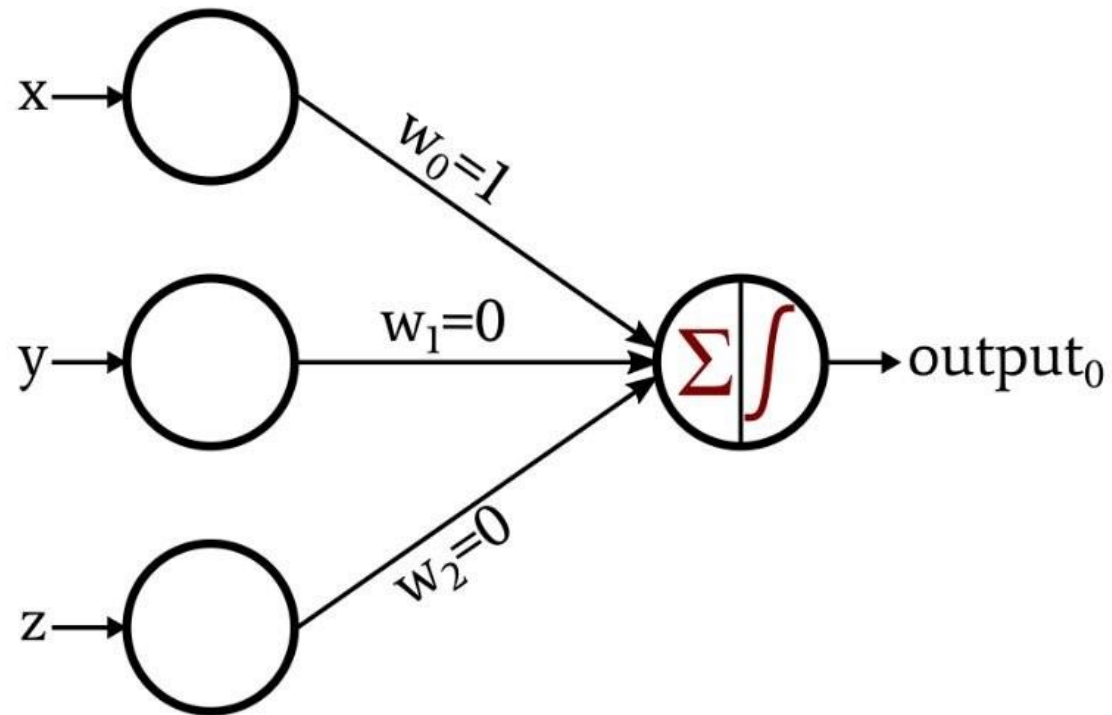


Limitaciones de la Neurona M-P

- Recibe únicamente valores binarios $\{1,0\}$, en la mayoría de los problemas reales se dispondrá de valores de otros tipos.
- Requiere la selección del threshold de manera manual.
- Todas las entradas son iguales. No se le puede asignar un mayor peso a una de las entradas.
- No son capaces de resolver problemas que no sean linealmente separables.

PERCEPTRON

- El perceptrón es una de las arquitecturas RNA más sencillas.
- Inventado en 1957 por Frank Rosenblatt.
- Se basa en un tipo de neurona artificial conocida como Threshold Logic Unit (TLU).
- Tanto las entradas como las salidas de la TLU se corresponden con valores numéricos continuos.

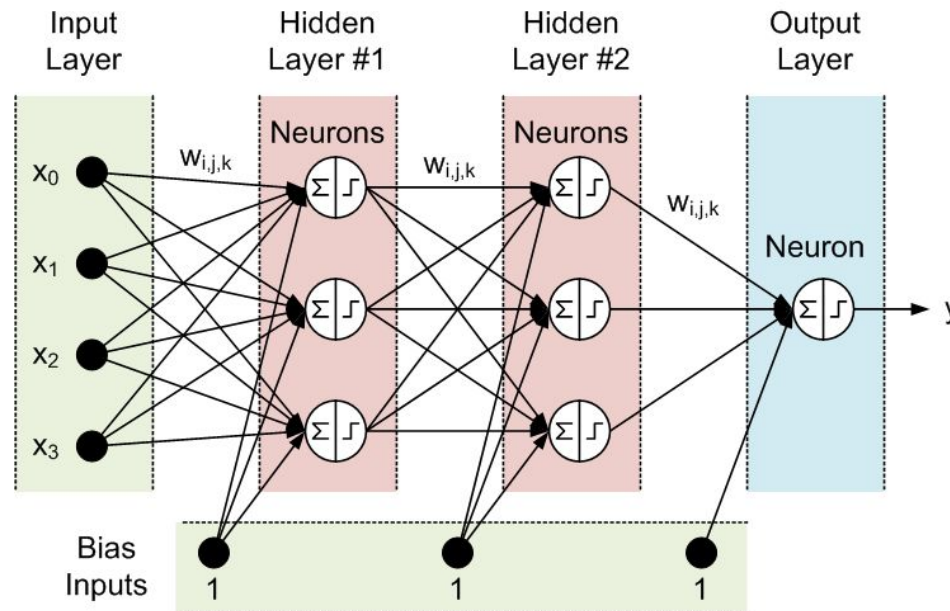


LIMITACIONES DEL PERCEPTRON

- El Perceptrón no proporciona como resultado una probabilidad, sino que realiza las predicciones basándose en un threshold estático.
- Construye límites de decisión lineales.
- El uso de Perceptrones tiene grandes limitaciones, no permite resolver problemas sencillos como el problema de clasificación XOR.
- Más adelante, se descubre que muchas de las limitaciones del Perceptrón se solucionan añadiendo más capas de TLUs, apareciendo el Multi-Layer Perceptron.

PERCEPTRÓN MULTICAPA

- El perceptrón Multicapa se compone de una input layer y una o más capas de TLUs.
- Cada capa de TLUs intermedia se denomina hidden layer.
- La capa final de TLUs es la output layer.
- Todas las capas excepto la output layer incluyen la bias neuron.
- Todas las capas se encuentran totalmente conectadas (fully connected) con las siguientes capas.
- Cuando una ANN tiene dos o más hidden layers, se denomina Deep Neural Network (DNN).



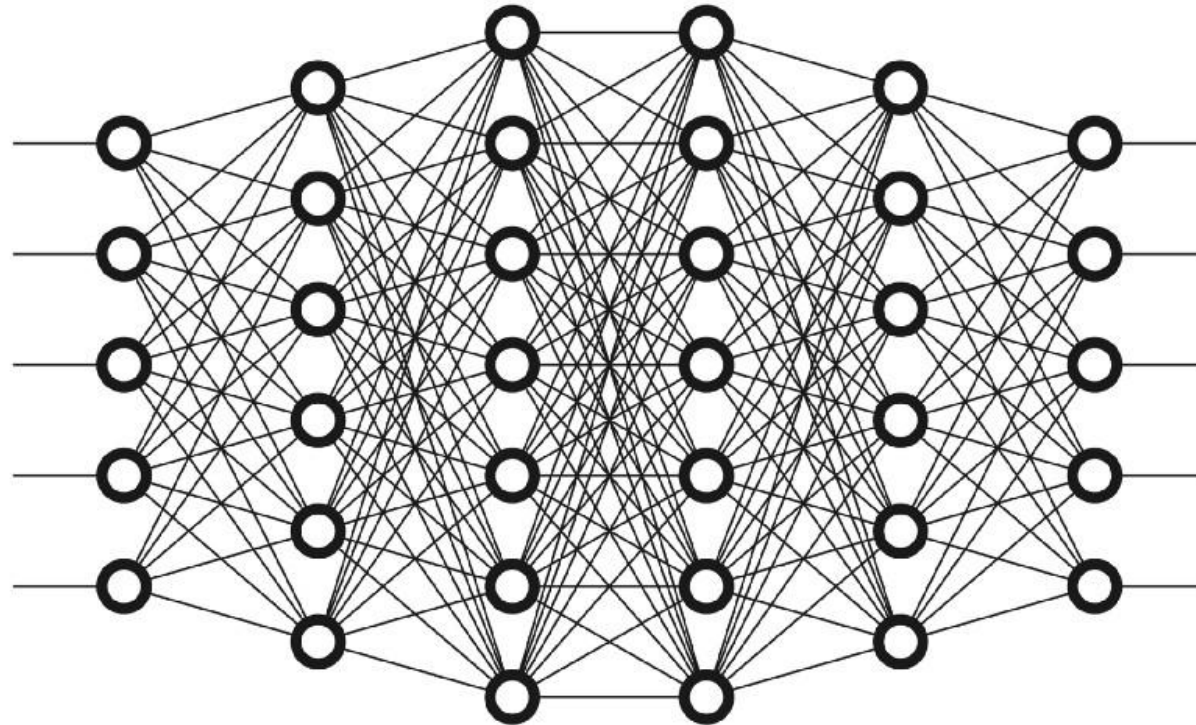
PERCEPTRÓN MULTICAPA

- Durante mucho tiempo una de las mayores limitación del Perceptrón Multicapa era que no existía una forma adecuada de entrenarlo.
- En 1986, D. E. Rumelhart presenta un algoritmo que revoluciona la manera de entrenar el Perceptrón Multicapa, el algoritmo backpropagation.
- El cambio clave del algoritmo backpropagation respecto a los utilizados anteriormente fue reemplazar la función de activación Heaviside step function por la sigmoide.
- En la actualidad existen otras funciones de activación populares, como la $\tanh(z)$ o la $\text{ReLU}(z)$.

TIPOS DE REDES NEURONALES

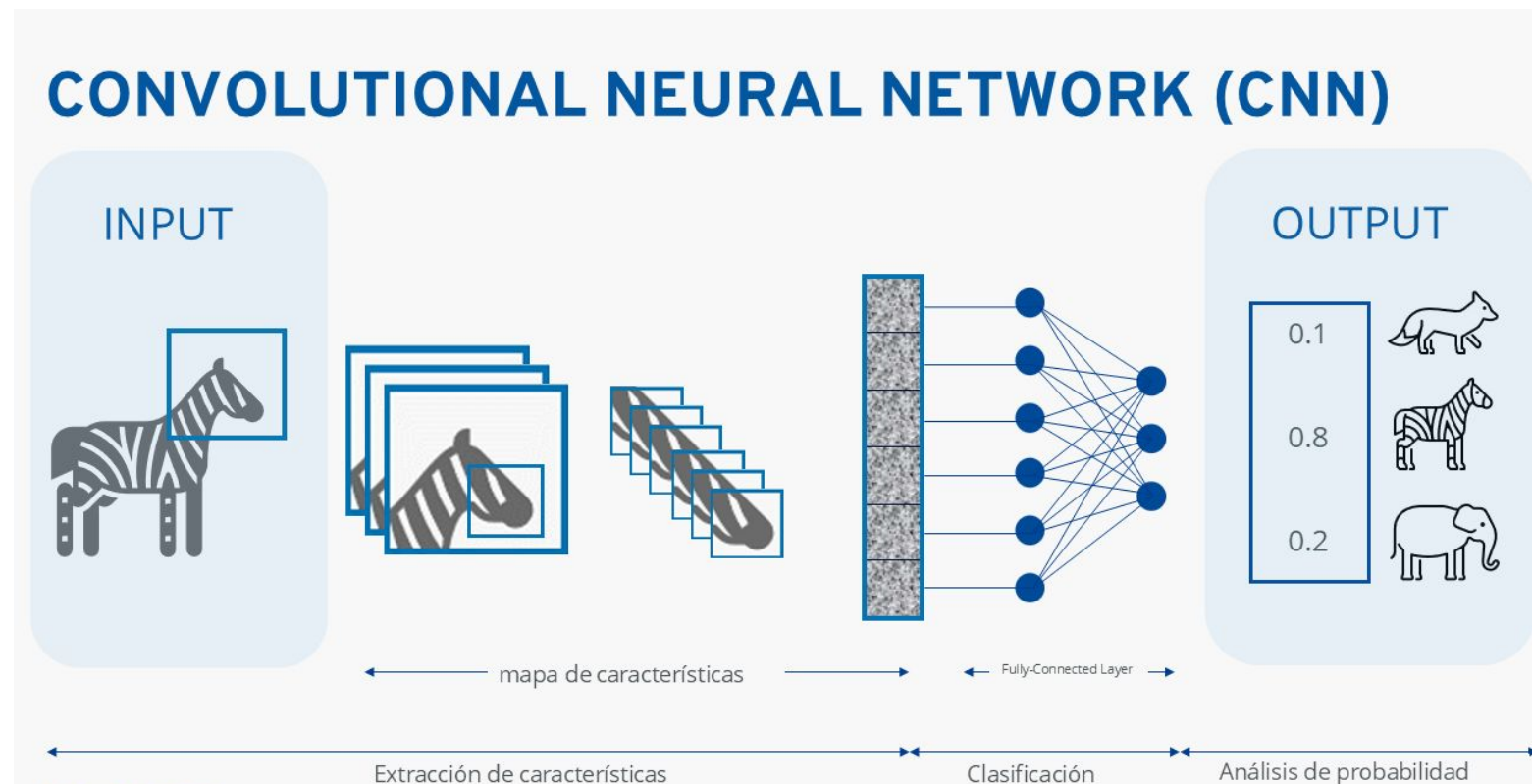
REDES NEURONALES FULLY CONNECTED

Este tipo de redes neuronales nos sirve para resolver problemas de clasificación por ejemplo en imágenes, de categorías como malicioso o legítimo, en transacciones fraudulentas, spam y en general la mayoría de temas previamente vistos en el curso.



REDES NEURONALES CONVOLUCIONALES

Este tipo de redes se suelen utilizarse para imágenes, estas no se conectan todas con todas, sus casos de uso son la conducción autónoma donde se reconoce fragmentos de imágenes continuas (fragmentos de un video).



CONVOLUCIÓN 2D

- La matriz representa el filtro a aplicar
- Aplicamos el filtro a cada uno de los píxeles de la imagen
- Se estila a tener la imagen de grises

| | | | |
|---|---|---|---|
| 2 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 3 | 0 | 0 |

*An input image
(no padding)*

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |

*A filter
(3x3)*

| | |
|--|--|
| | |
| | |

*Output image
(after convolving with stride 1)*

CONVOLUCIÓN 2D

| | | | |
|---|---|---|---|
| 2 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 3 | 0 | 0 |

*An input image
(no padding)*



| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |

*A filter
(3x3)*



| | |
|---|--|
| 3 | |
| | |

*Output image
(after convolving with stride 1)*

| | | | |
|---|---|---|---|
| 2 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 3 | 0 | 0 |

*An input image
(no padding)*



| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |

*A filter
(3x3)*



| | |
|---|---|
| 3 | 2 |
| | |

*Output image
(after convolving with stride 1)*

CONVOLUCIÓN 2D

| | | | |
|---|---|---|---|
| 2 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 3 | 0 | 0 |

*An input image
(no padding)*



| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |

*A filter
(3x3)*



| | |
|---|---|
| 3 | 2 |
| 3 | |

*Output image
(after convolving with stride 1)*

| | | | |
|---|---|---|---|
| 2 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 3 | 0 | 0 |

*An input image
(no padding)*



| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |

*A filter
(3x3)*



| | |
|---|---|
| 3 | 2 |
| 3 | 1 |

*Output image
(after convolving with stride 1)*

CONVOLUCIÓN 2D

- La matriz representa el filtro a aplicar
- Aplicamos el filtro a cada uno de los píxeles de la imagen
- Se estila a tener la imagen de grises



| | | |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |



EJEMPLO

filter2D : Aplica una convolución a una imagen haciendo uso de un kernel

Sintaxis: cv2.filter2D(imagen, profundidad, kernel)

Parametros:

imagen : Imagen a la cual aplicar la convolución

profundidad: Es la profundidad deseada para la imagen resultante

kernel: Filtro a aplicar en la imagen. En el caso que la imagen tenga varios canales, se puede dividir la imagen en sus distintos canales para aplicar diferentes filtros en cada canal. Los valores deben estar como punto flotante

Nota La profundidad es la precisión que tiene cada píxel. Define la cantidad de bits a usar y si se usa «unsigned, int, float o double». Si es igual a -1, entonces la imagen resultante tiene la misma precisión que la imagen original